


```
DDDDDDDD  IIIIII  SSSSSSSS  KK  KK  QQQQQQ
DDDDDDDD  IIIIII  SSSSSSSS  KK  KK  QQQQQQ
DD  DD  II  SS  SS  KK  KK  QQ  QQ
DD  DD  II  SS  SS  KK  KK  QQ  QQ
DD  DD  II  SS  SS  KK  KK  QQ  QQ
DD  DD  II  SS  SS  KK  KK  QQ  QQ
DD  DD  II  SS  SS  KK  KK  QQ  QQ
DD  DD  II  SS  SS  KK  KK  QQ  QQ
DD  DD  II  SS  SS  KK  KK  QQ  QQ
DD  DD  II  SS  SS  KK  KK  QQ  QQ
DD  DD  II  SS  SS  KK  KK  QQ  QQ
DDDDDDDD  IIIIII  SSSSSSSS  KK  KK  QQQQ  QQ
DDDDDDDD  IIIIII  SSSSSSSS  KK  KK  QQQQ  QQ
```

```
LL  IIIIII  SSSSSSSS
LL  IIIIII  SSSSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SSSSSS
LL  II  SSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```


(2)	46	DEFINITIONS
(3)	93	OWN STORAGE
(4)	129	Main routine
(5)	167	Create list of mounted disks
(6)	203	Scan the disk queues
(6)	251	Display the disk queue data


```
0000 1 .TITLE DISKQ Display Sizes of Disk I/O Queues
0000 2 .IDENT 'V04-000'
0000 3
0000 4 *****
0000 5
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *****
0000 26
0000 27 AUTHOR:
0000 28
0000 29 Len Kawell
0000 30
0000 31 MODIFIED BY:
0000 32
0000 33 V03-003 TCM0002 Trudy C. Matthews 09-Jun-1983
0000 34 Fix bug in TCM0001.
0000 35
0000 36 V03-002 TCM0001 Trudy C. Matthews 23-May-1983
0000 37 Set up inputs to IOC$CVT_DEVNAM so that node name is
0000 38 returned if the device is on a different node, and just
0000 39 the device name is returned if the device is local.
0000 40
0000 41 V03-001 KDM0002 Kathleen D. Morse 28-Jun-1982
0000 42 Added $DCDEF and $DEVDEF.
0000 43
0000 44 :---
```



```
0000 46 .SBTTL DEFINITIONS
0000 47 :
0000 48 : PUT_SCREEN - MACRO TO FORMAT AND PUT A MESSAGE TO THE TERMINAL SCREEN
0000 49 :
0000 50 .MACRO PUT_SCREEN TEXT,LINE=#1,COL=#1,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,AR
0000 51 .SAVE LSB
0000 52 .PSECT PURE
0000 53 $$DESC=.
0000 54 .ASCID TEXT
0000 55 .RESTORE
0000 56
0000 57 .IF NB ARG1 ; IF FORMATTING NEEDED
0000 58 MOVAB -128(SP),SP ; ALLOCATE FORMAT BUFFER
0000 59 PUSHL SP ; CREATE BUFFER DESCRIPTOR
0000 60 PUSHL #128 ;
0000 61 MOVL SP,R0 ; GET ADDR OF DESCRIPTOR
0000 62 $FAO_S $$DESC,(R0),(R0),- ; FORMAT THE OUTPUT
0000 63 ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8
0000 64 PUSHL COL ; PUSH COLUMN NUMBER
0000 65 PUSHL LINE ; PUSH LINE NUMBER
0000 66 PUSHAB 8(SP) ; SET ADDR OF BUFFER DESC
0000 67 CALLS #3,G^SCR$PUT_SCREEN ; OUTPUT THE FORMATTED TEXT
0000 68 MOVAB 128+8(SP),SP ; DEALLOCATE BUFFER AND DESC
0000 69
0000 70 .IFF
0000 71 PUSHL COL ; PUSH COLUMN NUMBER
0000 72 PUSHL LINE ; PUSH LINE NUMBER
0000 73 PUSHAB $$DESC ; SET ADDR OF TEXT DESC
0000 74 CALLS #3,G^SCR$PUT_SCREEN ; OUTPUT THE TEXT
0000 75 .ENDC
0000 76
0000 77 .ENDM PUT_SCREEN
0000 78
0000 79 :
0000 80 : Equated Symbols
0000 81 :
0000 82 $AQBDEF ; ACP queue block definitions
0000 83 $DCDEF ; Adapter type definitions
0000 84 $DDBDEF ; Device data block definitions
0000 85 $DEVDEF ; Device type definitions
0000 86 $IRPDEF ; I/O request packet definitions
0000 87 $UCBDEF ; Unit control block definition
0000 88 $VCBDEF ; Volume control block definitions
0000 89
00000010 0000 90 MAX_DISKS = 16 ; Maximum number of disks
00000010 0000 91 NAME_SIZ = 16 ; Size of disk name buffer
```



```
0000 93 .SBTTL OWN STORAGE
0000 94 :
0000 95 : Own Storage
0000 96 :
0000 97 NAME_LIST: ; Disk device names
00000100 0000 98 .BLKB MAX_DISKS*NAME_SIZ
00000140 0100 99 UCB_LIST: ; Disk UCB addresses
00000140 0100 100 .BLKL MAX_DISKS
00000144 0140 101 DISK_COUNT: ; Number of disks
00000144 0140 102 .BLKL 1
0144 103
0144 104 WAIT_TIME: ; Wait time between scans
FFFFFFFF FFF0BDC0 0144 105 .LONG -1000*1000,-1 ; (.1 seconds)
014C 106 HEADER: ; Display header
014C 107 .ASCID " I/O Requests Average I/O ACP Requests"<10><13>-

52 20 4F 2F 49 09 00000154 '010E0000' 014C 108 " Pending Size Pending"<10><13>-
72 65 76 41 09 73 74 73 65 75 71 65 015A
20 50 43 41 09 4F 2F 49 20 65 67 61 0166
09 67 6E 69 64 6E 65 50 20 20 09 0D 0172
50 20 20 09 09 65 7A 69 53 20 20 20 017B
0D 0A 67 6E 69 64 6E 65 0187
0193
019B 109
019B 110 :
019B 111 : These arrays hold the numbers currently shown in the display
019B 112 :
019B 113
000001DB 019B 114 DSP_IO_REQ: .BLKL MAX_DISKS ; Number of disk I/O requests
0000021B 01DB 115 DSP_ACP_REQ: .BLKL MAX_DISKS ; Number of ACP requests
021B 116
021B 117 :
021B 118 : These arrays hold the counts from the latest sample
021B 119 :
0000025B 021B 120 IO_REQ: .BLKL MAX_DISKS ; Number of disk I/O requests
0000029B 025B 121 ACP_REQ: .BLKL MAX_DISKS ; Number of ACP requests
000002DB 029B 122 IO_REQ_SIZ: .BLKL MAX_DISKS ; Size of the I/O requests
0000031B 02DB 123 IO_REQ_TOT: .BLKL MAX_DISKS ; Total number of disk requests
031B 124
031B 125
031B 126 .DEFAULT DISPLACEMENT,WORD
031B 127
```



```
031B 129 .SBTTL Main routine
031B 130
0000 031B 131 START: .WORD 0
78 50 E9 031D 132 $CMKRNLS CREATE_LIST ; Create list of UCB's
032A 133 BLBC -R0,EXIT ;
032D 134 ;
032D 135 ; Display disk names
032D 136 ;
01 DD 032D 137 PUSHL #1 ; Set cursor to 1,1
01 DD 032F 138 PUSHL #1 ; and
00000000'GF 02 FB 0331 139 CALLS #2,G^SCR$ERASE_PAGE ; erase the screen
FE10 CF 0338 140 PUSHAQ HEADER ; Set address of header desc
00000000'GF 01 FB 033C 141 CALLS #1,G^SCR$PUT_SCREEN ; Display it
53 D4 0343 142 CLRL R3 ; Init disk count
50 53 10 C5 0345 143 10$: MULL3 #NAME_SIZ,R3,R0 ; Compute name list index
FCB3 CF40 9F 0349 144 PUSHAB NAME_LIST+1[R0] ; Set address of name string
7E FCAD CF40 9A 034E 145 MOVZBL NAME_LIST[R0],-(SP) ; Set size of name string
01 DD 0354 146 PUSHL #1 ; Set column number
7E 53 04 C1 0356 147 ADDL3 #4,R3,-(SP) ; Set line number
08 AE 9F 035A 148 PUSHAB 8(SP) ; Set addr of descriptor
00000000'GF 03 FB 035D 149 CALLS #3,G^SCR$PUT_SCREEN ; Display the disk name
FE31 CF43 01 CE 0364 150 MNEGL #1,DSP_IO_REQ[R3] ; Indicate nothing on screen
FE6B CF43 01 CE 036A 151 MNEGL #1,DSP_ACP_REQ[R3]
CF 53 FDCC CF F2 0370 152 AOBLS DISK_COUNT,R3,10$ ; Increment count and loop
0376 153 ;
0376 154 ; Scan queues, wait a while, and do it again
0376 155 ;
0376 156 MAIN_LOOP:
0376 157 $CMEXEC,S SCAN_QUEUES ; Scan the queues
04AC'CF 1F 50 E9 0383 158 BLBC -R0,EXIT
00 FB 0386 159 CALLS #0,DISPLAY_QUEUES ; Display the data
038B 160 $SETIMR,S DAYTIM=WAIT_TIME,EFN=#3 ; Set a timer
039A 161 $WAITFR,S EFN=#3 ; Wait a while
D1 11 03A3 162 BRB MAIN_LOOP ; Do it again
03A5 163 EXIT:
04 03A5 164 RET
03A6 165
```



```
.SBTTL Create List of mounted disks
03A6 167
03A6 168
03A6 169 CREATE_LIST:
03A6 170 .WORD ^M<R2,R3,R4,R5>
54 00000000'GF D0 03A8 171 MOVL G^SCH$GL_CURPCB,R4 ; Get current PCB address
00000000'GF 16 03AF 172 JSB G^SCH$IOLOCKR ; Lock the I/O database for read acc
53 D4 03B5 173 CLRL R3 ; Init device count
54 00000000'GF DE 03B7 174 MOVAL G^IOC$GL_DEVLIST,R4 ; Get address of DDB list
54 64 D0 03BE 175 10$: MOVL (R4),R4 ; Get address of next DDB
45 13 03C1 176 BEQL 40$ ; If EQL, done
55 04 A4 D0 03C3 177 MOVL DDB$L_UCB(R4),R5 ; Get address of first UCB
01 40 A5 91 03C7 178 CMPB UCB$B_DEVCLASS(R5),#DC$_DISK ; Is device a disk?
F1 12 03CB 179 BNEQ 10$ ; If NEQ no
2E 38 A5 13 E1 03CD 180 20$: BBC #DEV$V_MNT,UCB$L_DEVCHAR(R5),30$ ; Branch if not mounted
29 38 A5 18 E0 03D2 181 BBS #DEV$V_FOR,UCB$L_DEVCHAR(R5),30$ ; Branch if mounted foreign
FD23 CF43 55 D0 03D7 182 MOVL R5,UCB_LIST[R3] ; Save the UCB address
50 53 10 C5 03DD 183 MULL3 #NAME_SIZ,R3,R0 ; Compute name index
FC1A CF40 9F 03E1 184 PUSHAB NAME_LIST[R0] ; Compute name buffer address
50 0F D0 03E6 185 MOVL #NAME_SIZ-1,R0 ; Set buffer size
51 6E 01 C1 03E9 186 ADDL3 #1,(SP),R1 ; Set buffer address
54 54 DD 03ED 187 PUSHL R4 ; Save current DDB address
01 CE 03EF 188 MNEGL #1,R4 ; Get node name if external device
00000000'GF 16 03F2 189 JSB G^IOC$CVT_DEVNAM ; Get the device name
54 8E D0 03F8 190 MOVL (SP)+,R4 ; Restore DDB address
9E 51 90 03FB 191 MOVBL R1,@(SP)+ ; Save name size
53 D6 03FE 192 INCL R3 ; Increment number found
55 30 A5 D0 0400 193 30$: MOVL UCB$L_LINK(R5),R5 ; Get address of next UCB
C7 12 0404 194 BNEQ 20$ ; If NEQ there is one
B6 11 0406 195 BRB 10$ ; Else, try next device
FD33 CF 53 D0 0408 196 40$: MOVL R3,DISK_COUNT ; Save count of disks found
54 00000000'GF D0 040D 197 MOVL G^SCH$GL_CURPCB,R4 ; Get current PCB address
00000000'GF 16 0414 198 JSB G^SCH$IOUNLOCK ; Unlock I/O data base
50 01 D0 041A 199 MOVL #1,R0 ; Set success
04 041D 200 RET
041E 201
```



```
041E 203 .SBTTL Scan the disk queues
041E 204 SCAN_QUEUES:
041E 205 .WORD ^M<R2,R3,R4,R5,R6,R7>
0420 206 CLRL R3 ; Init device count
55 FCD9 CF43 D0 207 10$: MOVL UCB_LIST[R3],R5 ; Get UCB address
0422 208 ;
0428 209 ; Total-up number of I/O requests queued to driver
0428 210 ;
0428 211 CLRL IO_REQ[R3] ; Init I/O request count
13 64 A5 08 E1 212 BBC #UCBSV_BSY,UCBSW_STS(R5),20$ ; If clear device not busy
FDE4 CF43 D6 213 INCL IO_REQ[R3] ; Count one for current I/O
50 58 A5 D0 214 MOVL UCB$$_IRP(R5),R0 ; Get addr of current I/O req
50 32 A0 3C 215 MOVZWL IRPSW_BCNT(R0),R0 ; Get transfer size
FE56 CF43 50 C0 216 ADDL R0,IO_REQ_SIZE[R3] ; Add to size total
56 4C A5 DE 217 20$: MOVAL UCB$$_IOQFL(R5),R6 ; Get addr of request listhead
57 56 D0 218 MOVL R6,R7 ; Save a copy of it
56 66 D0 219 30$: MOVL (R6),R6 ; Get addr of next request
57 56 D1 220 CMPL R6,R7 ; End of list?
14 FDC2 CF43 D1 221 BEQL 40$ ; If EQL yes - no more requests
11 1E 222 CMPL IO_REQ[R3],#20 ; Probably stuck? ////temp////
FDBA CF43 D6 223 BGEQU 40$ ; If GEQU yes ////temp////
50 32 A6 3C 224 INCL IO_REQ[R3] ; Increment request count
FE30 CF43 50 C0 225 MOVZWL IRPSW_BCNT(R6),R0 ; Get transfer size
DF 11 226 ADDL R0,IO_REQ_SIZE[R3] ; Add to size total
046B 227 BRB 30$ ; Try next one
046D 228 ;
046D 229 ; Total-up requests queued to ACP
046D 230 ;
56 34 A5 D0 231 40$: MOVL UCB$$_VCB(R5),R6 ; Get address of VCB
23 13 232 BEQL 60$ ; If EQL none - dismounted
56 10 A6 D0 233 MOVL VCB$$_AQB(R6),R6 ; Get address of request queue
FDDF CF43 D4 234 CLRL ACP_REQ[R3] ; Init ACP request count
57 56 D0 235 MOVL R6,R7 ; Get address of listhead
57 67 D0 236 50$: MOVL (R7),R7 ; Get address of next request
56 57 D1 237 CMPL R7,R6 ; End of list?
14 FDCF CF43 D1 238 BEQL 60$ ; If EQL yes - no more
07 1E 239 CMPL ACP_REQ[R3],#20 ; Probably stuck? ////temp////
FDC7 CF43 D6 240 BGEQU 60$ ; If GEQU yes ////temp////
E9 11 241 INCL ACP_REQ[R3] ; Increment ACP request count
0496 242 BRB 50$ ; Get next one
FE3C CF43 FD80 CF43 C0 243 60$: ADDL IO_REQ[R3],IO_REQ_TOT[R3] ; Add I/O requests to total
04 53 FC9D CF F2 244 AOBLS DISK_COUNT,R3,70$ ; Increment disk count and loop
50 01 D0 245 MOVL #1,R0 ; Set success
04A8 246 RET
04A9 247
FF76 31 248 70$: BRW 10$ ;
04AC 249
04AC 250 .SBTTL Display the disk queue data
04AC 251
04AC 252
04AC 253 DISPLAY_QUEUES:
003C 04AC .WORD ^M<R2,R3,R4,R5>
5E 00000200 8F C2 04AE 254
5E DD 04B5 255
00000200 8F DD 04B7 256
5E DD 04BD 257
SUBL #512,SP
PUSHL SP
PUSHL #512
PUSHL SP
```



```
00000000'GF 01 FB 04BF 260 CALLS #1,G^SCR$SET_BUFFER
53 D4 04C6 261 CLRL R3 ; Init disk counter
04C8 262
04C8 263 ; Update number of I/O requests for this device, if changed
04C8 264
10$: ADDL3 #4,R3,R5 ; Compute line number on screen
FCC6 CF43 55 53 04 C1 04C8 265 ; Anything changed since last time?
FD4A CF43 D1 04CC 266 ; If not, don't bother updating
7C 13 04D5 267 ; Write number of I/O's pending
54 FD89 CF43 D0 050D 268 ; Get I/O transfer size total
FDC3 CF43 D5 0513 270 ; Any I/O's?
06 13 0518 271 ; If EQL, no
54 FDBC CF43 C6 051A 272 ; Compute average I/O size
0520 273 20$: PUT_SCREEN <'!5UL'>,R5,#26,R4 ; Write average I/O size
0553 274
0553 275 ; Update number of ACP requests for this device, if changed
0553 276
FC7F CF43 FD03 CF43 D1 0553 277 30$: CMPL ACP_REQ[R3],DSP_ACP_REQ[R3]
36 13 055C 278 ;
055E 279 PUT_SCREEN <'!3UL'>,R5,#45,ACP_REQ[R3] ; Write number of ACP requests
0594 280
0594 281 ; All done with this device, go on to the next one
0594 282
FBFE CF43 FC82 CF43 D0 0594 283 40$: MOVL IO_REQ[R3],DSP_IO_REQ[R3] ; Remember what's on screen
FC35 CF43 FCB9 CF43 D0 059D 284 MOVL ACP_REQ[R3],DSP_ACP_REQ[R3]
OD 53 FB96 CF F2 05A6 285 AOBLS DISK_COUNT,R3,39$ ; Increment count and loop
00 DD 05AC 286 PUSHL #0
00000000'GF 01 FB 05AE 287 CALLS #1,G^SCR$PUT_BUFFER
50 01 D0 05B5 288 MOVL #1,R0 ; Set success
04 05B8 289 RET
FF0C 31 05B9 290 39$: BRW 10$ ; Branch helper
05BC 292
05BC 293 .END START
```


DISKQ
Symbol table

Display Sizes of Disk I/O Queues F 9

16-SEP-1984 02:21:47 VAX/VMS Macro V04-00
5-SEP-1984 04:36:49 [UTIL32.SRC]DISKQ.MAR;1

Page 8
(6)

\$\$DESC	= 00000018 R	03
\$\$T1	= 00000000	
\$\$T2	= 00000004	
ACP_REQ	0000025B R	01
CREATE_LIST	000003A6 R	01
DCS_DISK	= 00000001	
DDBSL_UCB	= 00000004	
DEVSV_FOR	= 00000018	
DEVSV_MNT	= 00000013	
DISK_COUNT	00000140 R	01
DISPLAY_QUEUES	000004AC R	01
DSP_ACP_REQ	000001DB R	01
DSP_IO_REQ	0000019B R	01
EXIT	000003A5 R	01
HEADER	0000014C R	01
IO\$CVT_DEVNAM	***** X	01
IO\$GL_DEVLIST	***** X	01
IO_REQ	0000021B R	01
IO_REQ_SIZ	0000029B R	01
IO_REQ_TOT	000002DB R	01
IRPSW_BCNT	= 00000032	
MAIN_COOP	00000376 R	01
MAX_DISKS	= 00000010	
NAME_LIST	00000000 R	01
NAME_SIZ	= 00000010	
SCAN_QUEUES	0000041E R	01
SCH\$GL_CURPCB	***** X	01
SCH\$IO\$CKR	***** X	01
SCH\$IO\$UNLOCK	***** X	01
SCR\$ERASE_PAGE	***** X	01
SCR\$PUT_BUFFER	***** X	01
SCR\$PUT_SCREEN	***** X	01
SCR\$SET_BUFFER	***** X	01
START	0000031B R	01
SYSS\$CMEXEC	***** GX	01
SYSS\$CMKRNL	***** GX	01
SYSS\$FAO	***** X	01
SYSS\$SETIMR	***** GX	01
SYSS\$WAITFR	***** GX	01
UCB\$B_DEVCLASS	= 00000040	
UCB\$B_DEVCHAR	= 00000038	
UCB\$B_IOQFL	= 0000004C	
UCB\$B_IRP	= 00000058	
UCB\$B_LINK	= 00000030	
UCB\$B_VCB	= 00000034	
UCB\$V_BSY	= 00000008	
UCB\$W_STS	= 00000064	
UCB_LIST	00000100 R	01
VCB\$B_AQB	= 00000010	
WAIT_TIME	00000144 R	01

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS :	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK :	000005BC (1468.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$ABSS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
PURE	00000024 (36.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.07	00:00:00.25
Command processing	138	00:00:00.61	00:00:02.06
Pass 1	304	00:00:10.28	00:00:13.69
Symbol table sort	0	00:00:01.48	00:00:01.48
Pass 2	70	00:00:01.89	00:00:02.06
Symbol table output	7	00:00:00.08	00:00:00.08
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	555	00:00:14.45	00:00:19.65

The working set limit was 1500 pages.
56667 bytes (111 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1030 non-local and 17 local symbols.
293 source lines were read in Pass 1, producing 15 object records in Pass 2.
25 pages of virtual memory were used to define 22 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	5
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	13
TOTALS (all libraries)	18

1150 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:DISKQ/OBJ=OBJ\$:DISKQ MSRC\$:DISKQ/UPDATE=(ENH\$:DISKQ)+EXECMLS/LIB

0429 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

